Clarification Request

References: , BTL Test Plan 14
3.14.2 Supports Writable Out_Of_Service Properties
3.16.2 Supports Writable Out_Of_Service Properties

Date of BTL-WG Response: 9-Feb-2017
☒All actions necessitated have been completed

Background:
Specified Tests 135.1-2013 - 7.3.1.1 - Out_Of_Service, Status_Flags, and Reliability Tests

Steps:

1. IF (Out_Of_Service is writable) THEN
   WRITE Out_Of_Service = TRUE
   ELSE
   MAKE (Out_Of_Service TRUE)
2. VERIFY Out_Of_Service = TRUE
3. VERIFY Status_Flags = (?, FALSE, ?, TRUE)
4. REPEAT X = (all values meeting the functional range requirements of 7.2.1) DO {
   WRITE Present_Value = X
   VERIFY Present_Value = X
   }
5. IF (Reliability is present and writable) THEN
   REPEAT X = (all values of the Reliability enumeration appropriate to the object type except
   NO_FAULT_DETECTED) DO {
   WRITE Reliability = X
   VERIFY Reliability = X
   VERIFY Status_Flags = (?, TRUE, ?, TRUE)
   WRITE Reliability = NO_FAULT_DETECTED
   VERIFY Reliability = NO_FAULT_DETECTED
   VERIFY Status_Flags = (?, FALSE, ?, TRUE)
   }
6. IF (Out_Of_Service is writable) THEN
   WRITE Out_Of_Service = FALSE
   ELSE
   MAKE (Out_Of_Service FALSE)
7. VERIFY Out_Of_Service = FALSE
8. VERIFY Status_Flags = (?, ?, ?, FALSE)

Problem: When this test is executed against a Multistate-Value object that has an active FAULT_STATE Fault Algorithm (according to clause 13.4.5)

Assumption:
Even while OUT-OF-SERVICE is TRUE; the FAULT_STATE Fault Algorithm is executed (with consequent behavior requirements as specified in Test 8.4.2)

Consequence:

If a value written in Teststep 4 is a member of the FAULT_VALUES then the Teststep 5 will fail while verifying VERIFY Reliability = NO_FAULT_DETECTED when X was written with the specific value MULTISTATE_FAULT.

If the lastvalue written in Teststep 4 is not a member of the FAULT_VALUES then the Teststep 5 will fail while verifying VERIFY Reliability = X when X was written with the specific value MULTISTATE_FAULT.

In either of the 2 possible cases, we will fail the test!

Therefore we conclude that the above assumption is false and the FAULT_STATE Fault Algorithm should not be executed while OUT-OF-SERVICE is true!

Questions:

1. If OUT-OF-SERVICE is TRUE, shall we inhibit the CHANGE_OF_STATE algorithm which is specified to put "MULTISTATE_FAULT" into Reliability property?
2. If OUT-OF-SERVICE as TRUE does not inhibit the CHANGE_OF_STATE algorithm, shall we modify the Test condition for "8.4.2 CHANGE_OF_STATE Tests" and for "8.5.2 CHANGE_OF_STATE Tests"?

Response:

The standard is still ambiguous for both questions. BTL-WG will review the tests. Because it is ambiguous, step 5 of 7.3.1.1 may be skipped.